Transfer Learning on Stack Exchange Tags

Fanming Dong
Statistics
Stanford University
fdong@stanford.edu

Shifan Mao
Chemical Engineering
Stanford University
shifan@stanford.edu

Weiqiang Zhu
Earth Science
Stanford University
zhuwq@stanford.edu

Abstract

In this project, we implement different machine learning methods and test their ability to transfer the knowledge learned on training categories (*e.g.* biology, cooking) to predict tags of the unseen "physics" category on Stack Exchange. We use the Glove word2vector model as the bridge of transfer leaning and try to train a model to understand the similarity of the meanings of the "title+content" and "tags", which may be transfered across different categories. We use unsupervised models like latent-Dirichlet Allocation and Tf-idf weighted neighbor search and supervised models such as logistic regression and recurrent neural network. The prediction results across different categories achieve reasonable F-1 scores, but they also show very strong bias towards the training categories.

1 Introduction

What can you learn about physics from studying biology? Stack Exchange is an online forum where people crowd source answers to hot questions. For easy navigation of questions and answers, the challenge asks people to label questions with correct tags. But what if it is a question we have never seen before?

Machine learning methods like deep neural networks have demonstrated high performance on single domain task for many natural language precessing problems. But transfer learning is still a challenging problem. Is is possible to transfer the knowledge trained on one category to another new category? Transfer learning aims to make use of valuable knowledge in a source domain to help model performance in a target domain which may have only few training samples.

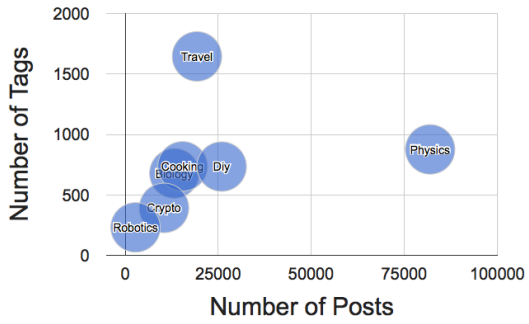
In this project, we used NLP methods to predict a posts tags on Stack Exchange according to its title and content. First, we train our models like tf-idf and RNN on single-domain tasks. Then, we explore these models transfer learning ability to predict tags across the six training categories and the unseen "physics" category.

2 Related Work

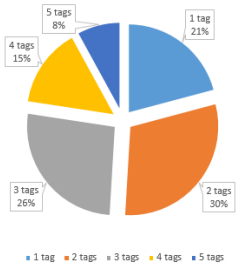
Collobert and Weston [1] applied multi-task learning in Natural Language Processing by training various tasks such as SRL, NER, POS, chunking and language modeling jointly. They demonstrated that learning tasks simultaneously can improve generalization performance. Luong et al. [2] examines three multi-task learning (MTL) settings for sequence to sequence models: one-to-many setting, many-to-one setting, and many-to-many setting. Liu et al. [3] used recurrent neural network for text classification with multi-task learning and proposed three different mechanisms for sharing information to model text with task-specific and shared layers.

	id	title	content	tags	category
0	1	What is the criticality of the ribosome bindin...	<p>In prokaryotic translation, how critical fo...	ribosome binding-sites translation synthetic-b...	biology
1	2	How is RNAse contamination in RNA based experi...	<p>Does anyone have any suggestions to prevent...	rna biochemistry	biology
2	3	Are lymphocyte sizes clustered in two groups?	<p>Tortora writes in Principles of Anatomy...	immunology cell-biology hematology	biology
3	4	How long does antibiotic-dosed LB maintain goo...	<p>Various people in our lab will prepare a li...	cell-culture	biology
4	5	Is exon order always preserved in splicing?	<p>Are there any cases in which the splicing m...	splicing mrna spliceosome introns exons	biology

Figure 1: Data Overview



(a) Relative distribution of number of posts against number of tags



(b) Number of tags of each post

Figure 2: Data exploration

3 Data Exploration

Our project uses dataset from a Kaggle competition (`https://www.kaggle.com/c/transfer-learning-on-stack-exchange-tags`). This dataset consists of tags and their corresponding title and content information from six domains: biology, cooking, crypto, diy, robotics, and travel. We are also given a test set which only includes title and content information of physics. Similar to most of the other Kaggle competitions, the tags on the test set are not given. Thus, we need to submit our work in order to find out how we do in the physics predictions.

3.1 Data Overview

From each of the six domains except for physics, we have title, content, and tags information. Figure 1 shows examples in our dataset.

Since we are interested in predicting physics tags, we have looked at the tags' distribution in each domain as well as the distribution and repetitive tags in the six domains as a whole. The things we have checked include number of unique tags, average number of tags per post, and etc (Figure 2). We have found that there are many unique tags and each post may have random number of tags associated with it. Also, we have found even though some tags appear in more than one domain, most of the tags do not.

3.2 Data Preprocessing

In order to implement our proposed models, we mainly do two preprocessing steps to the data: embedding and formatting. To facilitate further implementations, we considered word embeddings of both the Stack Exchange post texts (*i.e.* title and content) and tags. Specifically, we used GloVe

embeddings from Wikipedia 2014 + Gigaword 5 library [4]. Since we lack training set for Stack Exchange "physics" domain, word embedding becomes the only predictor for the tags.

In terms of formatting, we first stored the dataset in JSON format in order for a more standardized read and write process. Then, we write several functions to perform the following steps: pad the sequence in order to align the inputs, add tags to the end of either title or (title + content) input, add a dummy variable as response to indicate whether it is a true tag or a fake tag, and etc.

4 Approach

4.1 Deep Learning Related

1. Binary RNN model trained on title and "title + content"

In order to make this problem easier for transfer learning, we convert the predicting of tags to a binary classification problem. For each post ("title" and "content"), we go through every tags and predict whether this tag is true or false. Then, we train a binary RNN model to do the binary classification. This binary RNN model takes title (or "title + content") as input and append the true tags to the end of the input. In the case of tags such as "molecular-dynamics" we append both of them ("molecular", "dynamics") at the end of the input. Then, we give each post a "0/1" variable as a response indicating whether the tag is associated with the title. By training this binary RNN model, we can predict on the physics tags by running through each possible physics tag on each physics title (or "title + content") to see whether we should include the tag as an output. We use this F-1 score as a baseline to see if more advanced models perform better.

2. Binary RNN model with additional feature "if tags exist in tile/content"

We think one of the best guesses on whether a certain tag will be produced by a sentence is by checking if the tag has already appeared in the title (or content). This acts like an attention mechanism but do not involve training any attention vector seek relationship between tag with every word. The extra feature "if tags exist in tile/content" will be concatenated as a dummy variable to the output vector of RNN to indicate whether the tag has already shown up.

3. Binary RNN model with attention

In addition to using an indirect attention-like feature, we also try implementing a real attention model which has an attention vector that checks the relationship of the associated tags with every word in the title (or content).

4. Bidirectional Binary RNN model

We also try a bidirectional binary RNN model. By passing in title (or title + content) in both correct and reverse order, this may help RNN model to capture more information.

4.2 Unsupervised Learning Related

We found that a supervised trained model will result in bias when applied in a different domain. For instance, a trained model using subjects related to "biology" tend to predict tags with high relevance with biology when tested on "physics" dataset. Such bias is due to the intrinsic differences between Stack Exchange posts in different subjects. To overcome such biases, we resort to unsupervised learning models to discover appropriate tags for each post based on word embedding information.

1. Latent-Dirichlet Allocation

In our project, we seek to label tens of thousands posts with much fewer tags (on the order of hundreds for each subject). This implies that there are some common topics shared between different posts. In such scenario, topic discovery becomes a suitable method under an unsupervised setting. Latent-Dirichlet allocation (LDA) has been a popular model for topic discovery among large volumes of documents [5]. Inherently, LDA is a dimension reduction method aiming to cluster different documents based on the word frequencies information.

2. Tf-idf

In the first unsupervised learning model, we estimate the similarity between a post and all possible tags by summing over the similarities between a post's containing words of the tags. In specific, the similarity between a word in the post and the tag word is according to the cosine similarity of GloVe word embeddings. Additionally, the summation over word similarities is weighted by the tf-idf of words in a post.

5 Experiments

5.1 Supervised Single Domain Tag Prediction

In order to balance the portion of positive and negative samples for our binary classification, We have generated artificial negative samples during every epoch of training. The negative samples are generated by randomly selecting tags other than the true ones of the post. Because every post has only about three tags, there are a lot negative samples for prediction on the target "physics" domain. We need to choose what negative-to-positive ratio to use for training. In our work, we have tested the negative-to-positive from 1, 10, 30 to 100.

Then, we perform single domain classification. We apply data processing, training on training set, and evaluation on development set on each of the six categories separately to make sure our model makes sense. In the step of single domain classification, we run binary RNN model, binary RNN model with "tag-in-title/content" feature, binary RNN model with attention, as well as bidirectional binary RNN model. The corresponding results can be found in the results section below. We have also tested the effect of parameters such as negative-to-positive ratio, number of training epochs, training on "title" for "title+content". Due to the limited time and computation resources, we have not used grid search to fine tune these parameter.

5.2 Unsupervised Single Domain Tag Prediction

5.2.1 Topic discovery with latent Dirichlet allocation

Using LDA, we can first cluster similar posts into a certain number of topics. And then, the likelihood of having each tag can be estimated by measuring the similarity between the topic and the tag embedding. We measure the similarity between topic and tag by summing over the weighted averages of cosine similarities between word vectors and a tag vector. Mathematically, we write

$$L(\mathrm{tag}|\mathrm{post}) = \sum_{\mathrm{topic}} w(\mathrm{post}, \mathrm{topic})\mathrm{sim}(\mathrm{tag}, \mathrm{topic}) \tag{1}$$

where tag-topic-similarity is calculated from

$$\mathrm{sim}(\mathrm{tag}, \mathrm{topic}) = \sum_{\mathrm{word}} w(\mathrm{topic}, \mathrm{word}) \cos(\vec{u}_{\mathrm{word}}, \vec{u}_{\mathrm{tag}}). \tag{2}$$

In the above equations, the weighting functions $w(\mathrm{post}, \mathrm{topic})$ and $w(\mathrm{topic}, \mathrm{word})$ are weighting functions that were learned from LDA analysis.

Once we obtained the likelihood of a post having a particular tag, we can then select the most likely tags followed by the procedure of final tag selection described below.

In principle, this method first reduces the dimension of each post by embedding the titles into a low-dimension topic space. And then, appropriate tags can be applied to each post in such low-dimension space.

We face a major challenge of applying LDA in our project, primarily because we attempt to differentiate different posts in a common topic (*e.g.* physics). Thus it becomes hard to distinguish different topics when the topics have significant overlaps between each other. To overcome this difficulty, we first remove high-frequency words that commonly have high topic-word weighting $w(\mathrm{topic}, \mathrm{word})$, before calculating the post-tag likelihood according to Eq. 1.

5.2.2 tf-idf based embedding similarity

Besides topic modeling from a corpus of posts, we also considered the similarity between a post and each tag by directly looking at tf-idf weighted word embeddings and the tag embedding. This can be written as

$$P(\text{tag}|\text{post}) \propto \cos(\vec{u}_{\text{tag}}, \vec{u}_{\text{post}}). \tag{3}$$

This time the vector representation of each post is calculated from tf-idf weighted word embeddings

$$\vec{u}_{\text{post}} = \sum_{\text{word}} \text{tf} \cdot \text{idf}(\text{word}, \text{corpus}) \vec{u}_{\text{word}} \tag{4}$$

In particular, we used cosine similarity to measure word-to-word similarity to avoid the unwanted effect of highly-frequent words with large norms in Euclidean space.

5.3 Transfer Learning

Transfer learning aims to transfer the knowledge learned from a different task to make predictions for a new task. In this project, our main goal is to train our model on categories like biology to predict the correct tags of physics. The RNN models are good at understanding the meaning of sentences, which take one sentence as an input and output one vector. We are trying to transfer the model's trained ability to understand the sentence which is represented by the final vector and decide if the meaning of the tag matches the meaning of this sentence.

Word2vector models present the semantic and syntactic meanings and the relationship of words. In this project, we use the Glove model [4] to present the meaning of words in sentences and tags. We think the ability to determine if the meaning of sentences and tags are similar can be transfered by fixed embedding by GloVe model.

In order to verify the ability of transfer learning based on fixed GloVe word2vector model, we perform transfer learning across the six categories. For each categories, we trained the model using the training set of "title"s, then use this RNN model with the additional "tag-in-title" feature to predict the true tags of the other categories. We use 30:1 negative to positive ratio to generate the artificial negative samples. The training process is ran for 10 epochs.

6 Results

Here are the model results. We used F-1 score to compare models. F-1 score is calculated by the below formula:

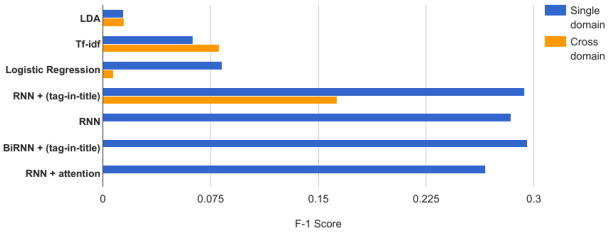
$$\text{F} - 1 \ \text{score} = 2pr/(p + r) \tag{5}$$

where precision $p = \text{tp}/(tp + fp)$ and recall $r = tp/(tp + fn)$. Number of true positive, false positive, and false negative samples are each denoted as tp, fp, and fn.

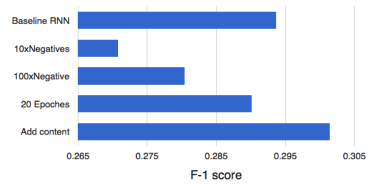
6.1 Supervised Single Domain Tag Prediction

On single domain, we choose category "biology" to test different models and parameters. For RNN models, we have tested five models: standard RNN model with concatenated titles and tags, RNN model with "tag-in-title" feature, BiRNN model with "tag-in-title" feature, and RNN model with attention. The F-1 scores of these five models on validation dataset are shown in 3a. Compared with standard RNN model, the F-1 score of RNN model with "tag-in-title" feature can be improved with little extra computation power. The BiRNN model has similar F-1 score as RNN model in this test. The RNN model with attention has an unexpected decease in F-1 score and needs much more computation time. So we have chosen the RNN model with "tag-in-title/content" feature for the transfer learning task.

Further, we tested different parameters based on one baseline RNN (3b). The baseline RNN model has "tag-in-title" feature. It is trained on "title" data of 30:1 negative-to-positive sample ratio and

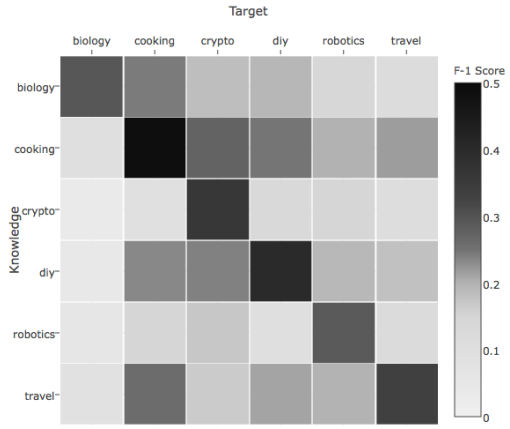


(a) F-1 scores of different models

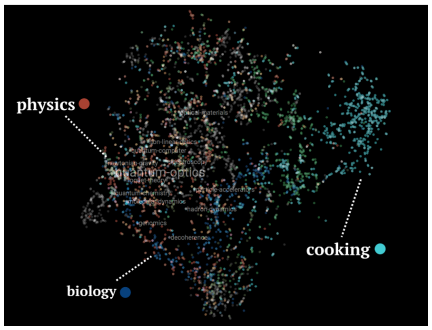


(b) F-1 scores of different RNN model parameters

Figure 3: F-1 scores of supervised single domain tag prediction



(a) F-1 score of cross-domain RNN models. Rows titled "knowledge" represent the RNN models trained on the labeled domain. The different columns are the target domains on which F-1 score are calculated.



(b) t-SNE reduced GloVe embeddings of tags from different domains.

runs for 10 epochs. Then, we tested different parameters independently which include 10:1 negative-to-positive ratio, 100:1 negative-to-positive ratio, 20 epochs and combined "title-content" inputs. The F-1 scores of these different parameter are shown in 3b. Due to limited computational power, we have not done grid search for parameter fine tuning. Based on the F-1 scores of these tested parameters, We choose to use both the baseline RNN models trained on "title" and "title and content" data for the transfer learning on physics.

6.2 Unsupervised Single Domain Tag Prediction

Since unsupervised learning does not rely on knowledge from other domains, the final predictions is only dependent on the titles and content of the posts within the domain. As expected, in an unseen domain "physics," we see similar performance of the model in terms of F-1 score. In fact, we achieved higher F-1 score in physics domain due to its larger dataset. Figure 3a shows that both LDA-based and Tf-idf based unsupervised learning have slight better performance when tested on physics domain.

6.3 Transfer Learning

Figure 4a displays the F-1 scores of single-domain and cross-domain models on the development set. Across the diagonal, we note that within a single domain, models have higher performance in less

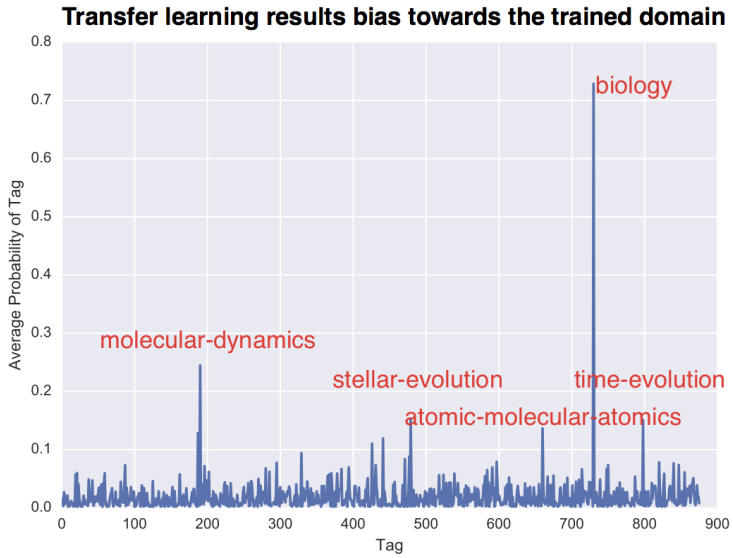


Figure 5: Total probability of tags of RNN predictions on physics domain trained on biology domain.

technical subjects. For instance, in categories named "cooking", "diy", and "travel", a baseline RNN model achieves F-1 scores greater than 0.30. For comparison, "biology", "robotics", and "crypto" can be less penetrable subjects.

Now we focus on the off-diagonal part of Figure 4a. The same trend above persists in cross-domain predictions. Using RNN models trained from other domains, biology remains to be a hard subject to predict tags on. Correspondingly, robotics is another difficult subject with low F-1 scores from cross-domain models. Surprisingly, crypto domain has modest F-1 scores even when predicted using cross-domain models.

Furthermore, because of the differences among disciplines, all F-1 scores are lower than within-domain predictions. Similarly, Figure 3a shows that an RNN model trained on biology gives significantly lower F-1 score when tested on an unseen domain physics, than within biology domain.

To investigate the performance of cross-domain models, we performed predictions on "physics" domain using RNN models trained on "biology" domain. The total probability of tags $\sum_{\text{post}} P(\text{tag}|\text{post})$ is shown in Figure 5. We observe large bias towards biology-related tags targeting physics tags. For example, the tag "biology" has very high probability although it doesn't exist in the biology category. Such bias can be argued by saying that RNN trained on biology is more likely to predict tags semantically similar to those in biology. Such observation is a key feature in our transfer learning. To resolve such bias, we resort to training the models on multiple domains, and taking extra measures in final tag selection to correct such bias.

6.4 Final tag selection

After we obtained the probabilities of each pair of post and tag, we follow a step of tag selection to improve the precision and recall of final predictions. A naive choice of tags chooses a common threshold of probabilities to identify the predicted tags for each post. We found the choice of threshold is highly sensitive to the ratio between positive and negative samples during model training (for supervised learning). Additionally, the choice of threshold tends to be large due to the very few numbers of correct tags for each post.

Therefore, we turn to a simple choice of top K tags with highest probabilities. On average, we found that a choice of top K = 4 most probable tags result in the highest F-1 score, balancing precision and recall. But this approach will give unrealistic uniform distribution of number of tags per post.

To incorporate the non-uniform distribution of number of tags per post (*e.g.* some multidisciplinary posts tend to have more tags than others), we look at the probability distribution of tags for a post. We found such probabilities take approximate form of Gaussian distribution. Therefore, it is reasonable to restrict our final prediction to the tails of the distribution with $P(\mathrm{tag}|\mathrm{post})$ greater than average probability of tags associated with a post. In the end, we take advantages of all above approaches and apply each selection method in sequence for final tag prediction.

An additional tag selection step is required for cross-domain learnings. Since we observe significant biased of tags towards trained-domain knowledge as we described above (see Fig. 5), we remove such bias by normalizing the probability of each tag $P(\mathrm{tag}|\mathrm{post})$, such that $\sum_{\mathrm{post}} P(\mathrm{tag}|\mathrm{post}) = 1$ for all tags. This is based on a naive assumption that each tag is equally likely. This assumption is evidently invalid as we observe distribution of popular tags in other domains. We will explore more rigorous correction of transfer-domain bias in the future.

7 Conclusions and Future Steps

In this paper, we have tested and analyzed effect of transfer leaning for tags prediction across different categories. We have conducted experiments of both unsupervised (LDA, Tf-idf) and supervised methods (Logistic regression and RNN models). The results show that RNN is much better than the other methods, which is known to be good at understanding the meaning of sentence. The performance of RNN on a single domain can achieve reasonable F-1 scores (0.48 for cooking), but the F-1 scores on transfer domains are much lower (0.25 at best). The prediction results are highly biased towards the domain of the training set. Unsupervised methods don't have the problem of bias term and have similar results on different categories. But it can only use the information of the training set which limits the F-1 score.

The transfer learning in this report is based on fixed Glove word2vector model. The effect of this transfer mechanism seems limited. How to design innovate RNN structures which can share and transfer common knowledge between related task needs further research. How to remove the bias term related to the training domain and only transfer the common knowledge is another direction for improvement.

During the project, we also find a new Topic-RNN model [6] which may be suitable to our task. As topic models focus on the global structure and RNN models capture local structure. Combining these two models may can give us more information on the more general structure and improve the prediction result on "physics" category.

References

[1] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[2] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

[3] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.

[4] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation, 2014.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[6] Adji B. Dieng, Chong Wang, Jianfeng Gao, and John William Paisley. Topicrnn: A recurrent neural network with long-range semantic dependency. *CoRR*, abs/1611.01702, 2016.